

Application No. 10/033,809

Responsive to Examiner Interview of Sept. 6, 2005

### **AMENDMENTS TO THE CLAIMS**

This listing of claims will replace all prior versions, and listings, of claims in the application:

#### **Listing of Claims:**

1. (Currently Amended) A computer-implemented method for bypassing I/O operations of a file system included in said computer, comprising:

including in said computer an operating system having application programming interfaces and a shell interface;

ordering computer code that includes I/O access commands in an application;

when said file system is optimized for processing queued I/O access commands that are a type of said I/O operations:

locating asynchronous direct said I/O access commands that are included in said application ordered computer code; and

bypassing said queued I/O access commands when porting said application from said operating system to a different operating system by executing said asynchronous direct I/O access commands by use of said application programming interfaces and said shell interface.

2. (Canceled)

Application No. 10/033,809

Responsive to Examiner Interview of Sept. 6, 2005

1 3. (Previously Presented) The computer-implemented method of Claim 1, further comprising  
2 bypassing said queued I/O access commands by use of a performance file.

1 4. (Currently Amended) A computer-implemented method for aggregating asynchronous direct  
2 I/O access commands, comprising:

3 including in said computer an operating system having application programming interfaces  
4 and a shell interface;

5 ordering computer code that includes I/O access commands in an application that does I/O  
6 caching;

7 supporting I/O request chaining in said computer that includes a file system;

8 when said file system is optimized for processing queued I/O access commands:

9 associating asynchronous direct I/O access commands with at least one file in said file  
10 system;

11 associating said at least one file with at least one performance file;

12 chaining said asynchronous direct I/O access commands into at least one aggregated  
13 I/O access command in said computer program application;

14 associating said at least one aggregated I/O access command with said at least one  
15 performance file;

16 identifying a terminus point in said ordered computer code;

17 issuing said at least one aggregated I/O access command by use of said application  
18 programming interfaces and said shell interface until said terminus point is

Application No. 10/033,809

Responsive to Examiner Interview of Sept. 6, 2005

19                   reached; and  
20                   when said terminus point is reached and if said at least one aggregated I/O command  
21                   remains, issuing a final said at least one aggregated I/O access command.

1     5. (Previously Presented) The computer-implemented method of Claim 4, further comprising:  
2         including data in said asynchronous direct I/O access commands; and  
3         including said data in said at least one aggregated I/O access command.

1     6. (Previously Presented) The computer-implemented method of Claim 4, further comprising  
2         allocating said performance file in single extents.

1     7. (Previously Presented) The computer-implemented method of Claim 4, further comprising  
2         pre-formatting said performance file.

1     8. (Previously Presented) The computer-implemented method of Claim 4, further comprising  
2         allocating said performance file in a named performance file pool.

1     9. (Previously Presented) The computer-implemented method of Claim 8, further comprising  
2         marking said performance file in said performance file pool as free.

1     10. (Previously Presented) The computer-implemented method of Claim 8, further comprising

Application No. 10/033,809

Responsive to Examiner Interview of Sept. 6, 2005

2 marking said performance file in said performance file pool as used.

1 11. (Previously Presented) The computer-implemented method of Claim 4, further comprising  
2 allocating said performance file in a default performance file pool.

1 12. (Previously Presented) The computer-implemented method of Claim 11, further comprising  
2 marking said performance file in said default performance file pool as free.

1 13. (Previously Presented) The computer-implemented method of Claim 11, further comprising  
2 marking said performance file in said default performance file pool as used.

1 14. (Previously Presented) The computer-implemented method of Claim 4, further comprising  
2 manipulating said performance file by a file pool utility.

1 15. (Original) The computer-implemented method of Claim 4, further comprising recovering  
2 from errors occurring while executing said at least one aggregated I/O access command.

1 16. (Previously Presented) The computer-implemented method of Claim 4, further comprising  
2 locating said asynchronous direct I/O access commands in a loop in said ordered computer code.

1 17. (Currently Amended) A computer system for bypassing I/O operations of a file system

Application No. 10/033,809

Responsive to Examiner Interview of Sept. 6, 2005

2 included in said computer system, comprising:

3 an operating system having application programming interfaces and a shell interface;

4 ordered computer code that includes I/O access commands in an application;

5 when said file system is optimized for processing queued said I/O access commands that are

6 a type of said I/O operations:

7 asynchronous direct I/O access commands that are included in said application

8 ordered computer code; and

9 said queued I/O access commands that are bypassed when said application is ported

10 from said operating system to a different operating system by executing said

11 asynchronous direct I/O access commands by use of said application

12 programming interfaces and said shell interface.

18. (Canceled)

1 19. (Previously Presented) The computer system of Claim 17, further comprising said queued

2 I/O access commands that are bypassed by use of a performance file.

1 20. (Previously Presented) A computer system for aggregating asynchronous direct I/O access

2 commands, comprising:

3 an operating system having application programming interfaces and a shell interface;

4 ordered computer code that includes I/O access commands in an application that does I/O

5 caching;

Application No. 10/033,809

Responsive to Examiner Interview of Sept. 6, 2005

6 a file system;  
7 when said file system is optimized for processing queued I/O access commands;  
8 asynchronous direct I/O access commands that are associated with at least one file in  
9 said file system;  
10 said at least one file that is associated with at least one performance file;  
11 said asynchronous direct I/O access commands that are chained into at least one  
12 aggregated I/O access command in said computer program application;  
13 a terminus point in said ordered computer code;  
14 said at least one aggregated I/O access command that is associated with said at least  
15 one performance file and that is issued until said terminus point is reached;  
16 and  
17 when said terminus point is reached and if said at least one aggregated I/O command  
18 remains, a final said at least one aggregated I/O access command.

1 21. (Previously Presented) The computer system of Claim 20, further comprising:  
2 data that is included in said asynchronous direct I/O access commands; and  
3 said data that is included in said at least one aggregated I/O access command.

1 22. (Previously Presented) The computer system of Claim 20, further comprising said  
2 performance file that is allocated in single extents.

Application No. 10/033,809

Responsive to Examiner Interview of Sept. 6, 2005

1 23. (Previously Presented) The computer system of Claim 20, further comprising said  
2 performance file that is a pre-formatted file.

1 24. (Previously Presented) The computer system of Claim 20, further comprising said  
2 performance file that is allocated in a named performance file pool.

1 25. (Previously Presented) The computer system of Claim 24, further comprising said  
2 performance file that is marked in said named performance file pool as free.

1 26. (Previously Presented) The computer system of Claim 24, further comprising said  
2 performance file that is marked in said named performance file pool as used.

1 27. (Previously Presented) The computer system of Claim 20, further comprising said  
2 performance file that is allocated in a default performance file pool.

1 28. (Previously Presented) The computer system of Claim 27, further comprising said  
2 performance file that is marked in said default performance file pool as free.

1 29. (Previously Presented) The computer system of Claim 27, further comprising said  
2 performance file that is marked in said default performance file pool as used.

Application No. 10/033,809

Responsive to Examiner Interview of Sept. 6, 2005

1 30.(Previously Presented) The computer system of Claim 20, further comprising said  
2 performance file that is manipulated by a file pool utility.

1 31. (Previously Presented) The computer system of Claim 20, further comprising said at least  
2 one aggregated I/O access command that recovers from errors.

1 32.(Previously Presented) The computer system of Claim 20, further comprising said  
2 asynchronous direct I/O access commands that are located in a loop in said ordered computer  
3 code.

1 33. (Canceled)

2

3 34.(Canceled)

4

5 35.(Canceled)

1 36. (Canceled)

1 37.(Canceled)



Application No. 10/033,809

Responsive to Examiner Interview of Sept. 6, 2005

1 38. (Currently Amended) An article of manufacture comprising a program storage medium  
2 readable by a computer and embodying one or more instructions executed by said computer for  
3 bypassing I/O operations of a file system included in said computer, wherein said article of  
4 manufacture is operable to:

5 include in said computer an operating system having application programming interfaces  
6 and a shell interface;  
7 order computer code that includes I/O access commands in an application;  
8 when said file system is optimized for processing queued I/O access commands that are a  
9 type of said I/O operations:  
10 locate asynchronous direct I/O access commands in said application ordered  
11 computer code; and  
12 bypass said queued I/O access commands when porting said application from said  
13 operating system to a different operating system by executing said  
14 asynchronous direct I/O access commands by use of said application  
15 programming interfaces and said shell interface.

1 39. (Previously Presented) The article of manufacture of Claim 38, further operable to bypass  
2 said queued I/O access commands by use of a performance file.

1 40. (Currently Amended) An article of manufacture comprising a program storage medium  
2 readable by a computer and embodying one or more instructions executed by said computer for

Application No. 10/033,809

Responsive to Examiner Interview of Sept. 6, 2005

3 aggregating asynchronous direct I/O access commands, wherein said article of manufacture is  
4 operable to:

5 include in said computer an operating system having application programming interfaces  
6 and a shell interface;

7 order computer code that includes I/O access commands in an application that does I/O  
8 caching;

9 support I/O request chaining in said computer that includes a file system;

10 when said file system is optimized for processing queued I/O access commands:

11 associate said asynchronous direct I/O access commands with at least one file in  
12 said file system;

13 associate said at least one file with at least one performance file;

14 chain said asynchronous direct I/O access commands into at least one aggregated  
15 I/O access command in said computer program application;

16 associate said at least one aggregated I/O access command with said at least one  
17 performance file;

18 identify a terminus point in said ordered computer code;

19 issue said at least one aggregated I/O access command by use of said application  
20 programming interfaces and said shell interface until said terminus point is  
21 reached; and

Application No. 10/033,809

Responsive to Examiner Interview of Sept. 6, 2005

- 22                    when said terminus point is reached and if said at least one aggregated I/O
- 23                    command remains, issue a final said at least one aggregated I/O access
- 24                    command.